

AMENDMENTS TO THE CLAIMS

1-27. (Canceled)

28. (Previously presented) A computer-implemented method for recreating a complex data object having a structure, the method comprising:

recording a sequence of calls from an authoring tool to a set of predefined functions used to create a complex data object;

translating the recorded sequence of calls into a sequence of directly executable instructions;

storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object;

obtaining the persistent representation of the structure of the complex data object and interpreting the directly executable instructions as calls to a set of predefined functions; and

calling a predefined function corresponding to each directly executable instruction in the sequence of directly executable instructions to construct the complex data object directly from the persistent representation.

29. (Previously presented) A method according to claim 28, further comprising displaying the complex data object to a user.

30. (Previously presented) A method according to claim 28, wherein the complex data object is a multimedia presentation.

31. (Previously presented) A method according to claim 28, wherein at least some of the functions have arguments.

32. (Previously presented) A method according to claim 31, wherein a call to one of the functions includes a call to another function as an argument of the first function.

33. (Previously presented) A method according to claim 31, wherein a call to one of the functions includes obtaining a constant value as its argument.

34. (Previously presented) A method according to claim 28, wherein at least some of the functions return an explicit result.

35. (Previously presented) A method according to claim 28, wherein the set of predefined functions further comprises information about the readable name of each function,

whether each function returns a result, whether each function has a side effect, and the number of arguments for each function.

36. (Previously presented) A method according to claim 35, wherein recording the sequence of calls further comprises obtaining the readable name of each function, whether each function returns a result, whether each function has a side effect, and the number of arguments for each function from the set of predefined functions.

37. (Previously presented) A method according to claim 36, wherein the sequence of calls are recorded as a sequence of code fragments representing each function called, the arguments to each function, and the order in which the authoring tool made the call.

38. (Previously presented) A method according to claim 28, wherein the sequence of directly executable instructions are interpreted by a stack-based virtual machine.

39. (Previously presented) A computer-readable medium having computer-executable instructions for performing the method recited in claim 28.

40. (Previously presented) A computer system having a processor, a memory, and an operating environment, the computer system being operable for performing the method recited in claim 28.

41. (Previously presented) A system for recreating a complex data object from a persistent representation of its structure, the system comprising:

- an authoring tool for recording a sequence of calls to a predefined set of data types and methods for creating a complex data object;

- a program generator for translating the recorded sequence of calls into a sequence of directly executable instructions and storing the sequence of directly executable instructions as a persistent representation of the structure of the complex data object;

- a library having a predefined set of data types and methods for creating complex data objects; and

- a program interpreter for interpreting the contents of the persistent representation as a sequence of directly executable instructions, and for executing those instructions as a sequence of calls on the library so as to construct the complex data object directly from the persistent representation.

42. (Previously presented) A system according to claim 41, where the complex data object is a multimedia presentation.

43. (Previously presented) A system according to claim 41, where the program interpreter is a virtual machine located in a computer in which the complex data object is presented to the user.

44. (Previously presented) A system according to claim 43, wherein the program interpreter is a stack-based virtual machine.

45. (Previously presented) A system according to claim 44, wherein the stack-based virtual machine further includes a temporary storage array.

46. (Previously presented) A system according to claim 41, wherein the predefined set of data types and methods further comprises information about the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method.

47. (Previously presented) A system according to claim 46, wherein recording the sequence of calls further comprises obtaining the readable name of each method, whether each method returns a result, whether each method has a side effect, and the number of arguments for each method from the predefined set of data types and methods.

48. (Previously presented) A system according to claim 47, wherein the sequence of calls are recorded as a sequence of code fragments representing each method called, the arguments to each method, and the order in which the authoring tool made the call.

49-65. (Canceled)